

IN THE CLAIMS:

1 1. (Currently amended) A method for avoiding starvation of interrupts of a com-
2 puter system, the computer system having at least one processor for servicing the inter-
3 rupts, a memory subsystem, one or more input/output (I/O) devices configured to issue
4 interrupts and at least one interrupt controller, the method comprising the steps of:
5 providing an interrupt queue register having a fixed number of entries for storing
6 interrupts to be serviced by the at least one processor;
7 posting interrupts generated by the one or more I/O devices to the interrupt queue
8 register;
9 determining whether the interrupt queue register is full and unable to store a given
10 ~~new~~ interrupt; and
11 if the interrupt queue register is full, one of waiting a preset time and reposting the
12 given ~~new~~ interrupt to the interrupt queue register, and draining the contents of the inter-
13 rupt queue register into the memory subsystem so that the interrupt queue register can
14 store the given ~~new~~ interrupt.

1 2. (Currently amended) The method of claim 1 further comprising the steps of:
2 if the interrupt queue register has a free entry for storing the given ~~new~~ interrupt,
3 sending an acknowledgement message (ACK) to the interrupt controller; and
4 if the interrupt queue register is full and cannot store the given ~~new~~ interrupt,
5 sending a non-acknowledgment message (NACK) to the interrupt controller.

1 3. (Original) The method of claim 2 further comprising the steps of:
2 providing an interrupt starvation counter that is operable by the interrupt control-
3 ler;
4 in response to receiving an ACK from the processor, clearing the interrupt starva-
5 tion counter; and

6 in response to receiving a NACK from the processor, adjusting the interrupt star-
7 vation counter.

1 4. (Original) The method of claim 3 wherein the step of adjusting comprises one
2 of incrementing or decrementing the interrupt starvation counter.

1 5. (Currently amended) The method of claim 3 further comprising the steps of:
2 determining whether the interrupt starvation counter exceeds a threshold; and
3 if the interrupt starvation counter does not exceed the threshold, performing the
4 step of waiting the preset time and reposting the given ~~new~~ interrupt to the interrupt
5 queue register.

1 6. (Original) The method of claim 5 further comprising the steps of:
2 providing an interrupt request register having a plurality of bit positions that are
3 settable to indicate a priority level of an interrupt posted to the processor;
4 if the interrupt starvation counter exceeds the threshold, setting the interrupt re-
5 quest register with a high priority indication; and
6 in response to the setting of the interrupt request register with the high priority
7 indication, performing the step of draining the contents of the interrupt queue register into
8 the memory subsystem.

1 7. (Original) The method of claim 6 further comprising the step of, in response to
2 the posting of an interrupt to the interrupt queue register, setting the interrupt request reg-
3 ister with a low priority indication.

1 8. (Original) The method of claim 7 wherein the interrupt request register includes
2 a first bit segment the setting of which is indicative of a low priority interrupt and a sec-
3 ond bit segment the setting of which is indicative of a high priority interrupt.

1 9. (Original) The method of claim 8 wherein each bit segment of the interrupt re-
2 quest register is associated with an interrupt line leading into the processor.

1 10. (Original) The method of claim 6 wherein the contents of the interrupt queue
2 register drained into the memory subsystem are placed in a first-in-first-out (FIFO)
3 queue.

1 11. (Currently amended) The method of claim 10 further comprising the step of
2 servicing the interrupts stored at the ~~in-memory~~ FIFO queue.

1 12. (Original) The method of claim 1 wherein the interrupts issued by the I/O de-
2 vices are one of level sensitive interrupts (LSIs) and message signaled interrupts (MSIs).

1 13. (Original) The method of claim 11 wherein the interrupts issued by the I/O
2 devices are one of level sensitive interrupts (LSIs) and message signaled interrupts
3 (MSIs).

1 14. (Original) The method of claim 1 wherein the step of draining the contents of
2 the interrupt queue register into the memory subsystem is performed by Privileged Archi-
3 tecture Library (PAL) code running at the processor.

1 15. (Original) The method of claim 1 wherein
2 the computer system includes a plurality of processors configured to issue inter-
3 rupts, and
4 the interrupt controller receives interrupts from one or more selected processors of
5 the computer system and posts such processor initiated interrupts to the interrupt queue of
6 the at least one processor as with the I/O device interrupts.

1 16. (Original) A computer system comprising:
2 a plurality of input/output (I/O) devices configured to issue interrupts;

3 at least one processor for servicing the interrupts;
4 a memory subsystem;
5 an interrupt controller configured to receive the interrupts from the I/O devices
6 and to post those interrupts to the at least one processor for servicing; and
7 an interrupt queue register for storing the interrupts posted to the at least one
8 processor, wherein
9 upon receiving a newly posted interrupt, the at least one processor determines
10 whether the interrupt queue register is already full and, if so, drains the contents of the
11 interrupt queue register into the memory subsystem.

1 17. (Original) The computer system of claim 16 further comprising an interrupt
2 starvation counter that is operable by the interrupt controller, wherein
3 if the interrupt queue register has a free entry for the new interrupt, sending an
4 acknowledgement (ACK) message from the at least one processor to the interrupt con-
5 troller, and
6 in response to the ACK, clearing the interrupt starvation counter.

1 18. (Original) The computer system of claim 17 further comprising
2 if the interrupt queue register does not have a free entry for the new interrupt,
3 sending a non-acknowledgement (NACK) message from the at least one processor to the
4 interrupt controller,
5 in response to the NACK, adjusting the interrupt starvation counter,
6 determining whether the interrupt starvation counter exceeds a threshold,
7 if the interrupt starvation counter does not exceed the threshold, waiting a prede-
8 termined time and reposting the new interrupt to the at least one processor.

1 19. (Original) The computer system of claim 18 further comprising an interrupt
2 request register having a plurality of bit positions that are settable to indicate a priority
3 level of an interrupt posted to the processor, wherein

4 if the interrupt starvation counter exceeds the threshold, setting the interrupt re-
5 quest register with a high priority indication; and
6 in response to the setting of the interrupt request register with the high priority
7 indication, draining the contents of the interrupt queue register into the memory subsys-
8 tem.

1 20. (Original) The computer system of claim 16 wherein
2 the computer system has a plurality of processors configured to issue interrupts,
3 and
4 the interrupt controller posts the processor initiated interrupts to the at least one
5 processor for servicing as with the I/O device interrupts.